

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

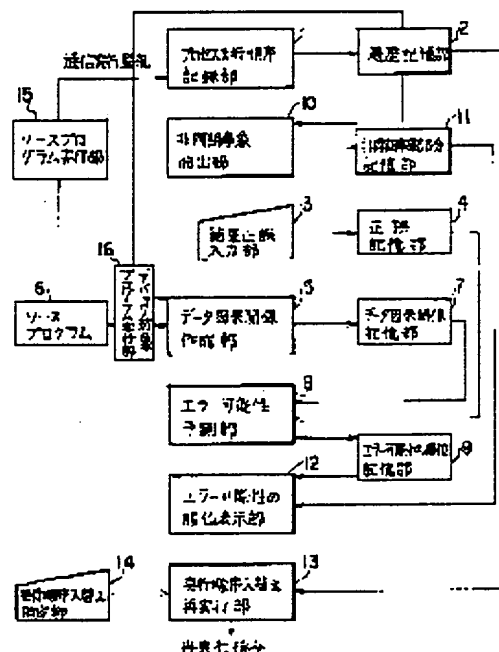
E 4934

(11)Publication number : 05-233323
(43)Date of publication of application : 10.09.1993

G06F 9/46
G06F 11/28

(71)Applicant : TOSHIBA CORP
(72)Inventor : KITAMURA ASAKO
HASEGAWA TETSUO
SEKI TOSHIKUMI

CONSTITUTION: A data causal relation generating part 5 which generates data causal relation information from relation between the data of the debug target process and the execution sequence among processes based on program execution sequence information among the processes obtained by the execution of a source program 6 and information obtained by the execution of the debug target process by the source program 6, and a display part 12 which displays the possibility of a program error predicted based on the data causal relation generated at the data causal relation generating part 5 and the program execution sequence among the processes as debug supporting information are provided at a computer system which performs processing as performing message communication among plural processes by the execution of the source program 6.



[Date of request for examination]	14.01.1999
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-233323

(43)公開日 平成5年(1993)9月10日

(51)IntCl.⁵

G 0 6 F 9/46
11/28

識別記号

3 3 0 C
A

庁内整理番号

8120-5B
9290-5B

F I

技術表示箇所

審査請求 未請求 請求項の数3(全 11 頁)

(21)出願番号 特願平4-35353

(22)出願日 平成4年(1992)2月21日

(71)出願人 000003078

株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者 北村 麻子

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(72)発明者 長谷川 哲夫

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(72)発明者 関 俊文

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

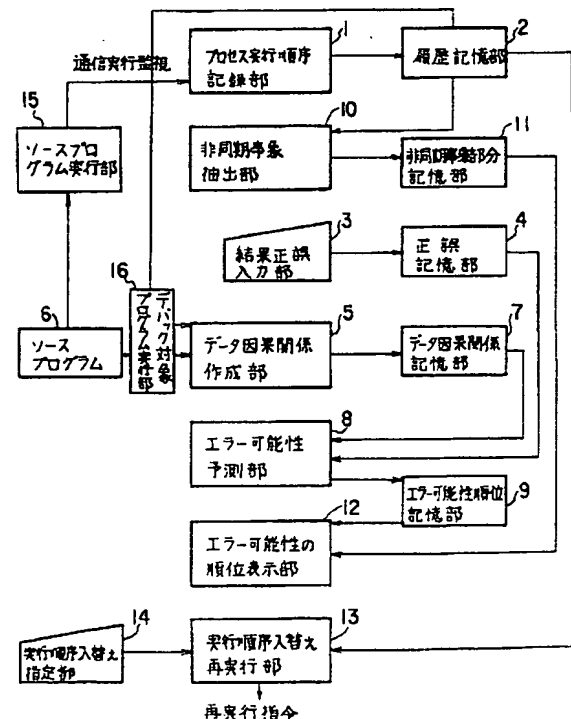
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 並行プログラムのデバッグ支援装置

(57)【要約】

【目的】デバッグ対象プロセスの非同期部分の全てに対して実行順序を入替えてテストすることなく、必要最小限の非同期事象の組合せでデバッグを対話的に効率良く行うことにある。

【構成】ソースプログラム6の実行により複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、ソースプログラム6の実行により得られるプロセス間のプログラム実行順序情報とソースプログラム6によるデバッグ対象プロセスの実行により得られる情報とに基づき前記デバッグ対象プロセスのデータと前記プロセス間の実行順序との関係からデータ因果関係情報を作成するデータ因果関係作成部5と、このデータ因果関係作成部5で作成されたデータ因果関係とプロセス間のプログラム実行順序に基づいて予測されるプログラムエラーの可能性をデバッグ支援情報として表示する表示部12とを備える。



【特許請求の範囲】

【請求項1】 ソースプログラムの実行により複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、前記ソースプログラムの実行により得られるプロセス間のプログラム実行順序情報を用いて前記ソースプログラムによるデバッグ対象プロセスの実行により得られる情報に基づき前記デバッグ対象プロセスのデータの定義およびデータ間の参照関係からデータの因果関係情報を作成するデータ因果関係作成手段と、このデータ因果関係作成手段で作成されたデータ因果関係に前記ソースプログラムの実行により得られた結果の正誤をもとに予測したデータの引数として受取る部分でのプログラムエラーの可能性をデバッグ支援情報として表示する表示手段とを備えたことを特徴とする並行プログラムのデバッグ支援装置。

【請求項2】 表示手段に表示されるデバッグ支援情報は、データ因果関係情報にプロセス間のプログラム実行順序から抽出された実行順序の入替え可能な非同期事象部分を合わせて表示可能な情報としたことを特徴とする請求項1に記載の並行プログラムのデバッグ支援装置。

【請求項3】 ソースプログラムの実行により複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、前記ソースプログラムの実行により得られるプロセス間のプログラム実行順序情報と前記ソースプログラムによるデバッグ対象プロセスの実行により得られる情報とに基づき前記デバッグ対象プロセスのデータと前記プロセス間のプログラム実行順序との関係からデータ因果関係情報を作成するデータ因果関係作成手段と、このデータ因果関係作成手段で作成されたデータ因果関係に前記ソースプログラムの実行により得られた結果をもとに予測したデータを引数として受取る部分でのプログラムエラーの可能性とプロセス間のプログラム実行順序に基づいて予測される実行順序の入替え可能な非同期事象部分を併せてデバッグ支援情報として表示する表示手段と、この表示手段に表示された表示内容に従って前記非同期事象部分に対する実行順序の入替えが指定されると前記プロセス間のプログラム実行順序情報をもとに前記デバッグ対象プロセスの非同期事象に対する実行順序を入替える再実行指令を出力する実行順序入替え再実行手段とを備えたことを特徴とする並行プログラムのデバッグ支援装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、デバッグの対象となるプロセス（以下デバッグ対象プロセスと呼ぶ）に存する実行順序の入替え可能な非同期事象部分で生じるプログラムエラーを分析、検出するための並行プログラムのデバッグ支援装置に関する。

【0002】

【従来の技術】 複数のプロセス間でメッセージ通信をしながらプログラム処理を進める計算機システムにおいて、プロセス中にメッセージに対する同期の取り忘れや、意図的に同期を取っていない部分（実行順序の入替可能な非同期事象部分：ここでは単に非同期事象部分と呼ぶ）があると、他のプロセスからのメッセージの到着する順番が変わった場合には、ユーザの意図しないプロセス間の通信によるプログラムエラーが発生することがある。従って、このプログラムエラーを検出するためには、予めプログラムエラーとなる非同期事象部分がどこであるかを検索しておく必要がある。

【0003】 従来、かかるプログラムエラーとなる非同期事象部分を検索するには、まずソースプログラムの実行により各プロセス間のプログラム実行順序を取込んで、デバッグ対象プロセス中に存する非同期事象部分を抽出し、次にデバッグ対象プロセスの非同期事象部分の全ての組合わせで実行順序を入替えてテストすることによりプログラムエラーとなる非同期事象部分を検索するようにしていた。

【0004】

【発明が解決しようとする課題】 しかし、このような検索手法では、全ての非同期事象部分に対して実行順序を入替えて再実行テストをしなければならず、しかもその組合わせ数も多数存在するため、デバッグに膨大な時間と手間がかかるという問題がある。

【0005】 本発明は、デバッグ対象プロセスの非同期事象部分の全てに対して実行順序を入替えてテストすることなく、必要最小限の非同期事象部分の組合わせでデバッグを対話的に効率良く行うことができる並行プログラムのデバッグ支援装置を提供することを目的とする。

【0006】

【課題を解決するための手段】 本発明は上記の目的を達成するため、ソースプログラムの実行により複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、前記ソースプログラムの実行により得られるプロセス間のプログラム実行順序情報と前記ソースプログラムによるデバッグ対象プロセスの実行により得られる情報とに基づき前記デバッグ対象プロセスのデータと前記プロセス間のプログラム実行順序との関係からデータの因果関係情報を作成するデータ因果関係作成手段と、このデータ因果関係作成手段で作成されたデータ因果関係とプロセス間のプログラム実行順序に基づいて予測される実行順序の入替え可能な非同期事象部分でのプログラムエラーの可能性をデバッグ支援情報として表示する表示手段とを備えたものである。

【0007】 また、ソースプログラムの実行により複数のプロセス間でメッセージ通信をしながら処理を進める計算機システムにおいて、前記ソースプログラムの実行により得られるプロセス間のプログラム実行順序情報と前記ソースプログラムによるデバッグ対象プロセスの非

行により得られる情報とに基づき前記デバッグ対象プロセスのデータと前記プロセス間のプログラム実行順序との関係からデータ因果関係情報を作成するデータ因果関係作成手段と、このデータ因果関係作成手段で作成されたデータ因果関係とプロセス間のプログラム実行順序に基づいて予測される実行順序の入替可能な非同期事象部分でのプログラムエラーの可能性をデバッグ支援情報として表示する表示手段と、この表示手段に表示された表示内容に従って前記非同期事象部分に対する実行順序の入替えが指定されると前記プロセス間のプログラム実行順序情報をもとに前記デバッグ対象プロセスの非同期事象に対する実行順序を入替える再実行指令を出力する実行順序入替え再実行手段とを備えたものである。

【0008】

【作用】このような構成の並行プログラムのデバッグ支援装置にあっては、データ因果関係作成手段により、ソースプログラムの実行により得られるプロセス間のプログラム実行順序情報とソースプログラムによるデバッグ対象プロセスの実行により得られる情報とに基づきデバッグ対象プロセスのデータとプロセス間の実行順序との関係からデータ因果関係情報が作成され、このデータ因果関係とプロセス間のプログラム実行順序に基づいて予測される実行順序の入替可能な非同期事象部分でのプログラムエラーの可能性がデバッグ支援情報として表示手段に表示されるので、プログラムエラーとなる実行順序の入替えが可能な非同期事象部分の検索が容易に行うことができる。

【0009】また、実行順序入替え再実行手段により表示手段に表示された表示内容に従ってプログラムエラーの可能性の高いデバッグ対象プロセスの非同期事象に対する入替えを行って再実行されるので、デバッグ対象プロセスの非同期部分の全てに対して実行順序を入替えてテストすることなく、必要最小限の非同期事象の組合せでデバッグ支援を対話的に効率良く行うことが可能となり、デバッグに要する時間および手間を大幅に削減することができる。

【0010】

【実施例】以下本発明の一実施例を図面を参照して説明する。

【0011】図1は本発明による並行プログラムのデバッグ支援装置の構成例を示すブロック図である。図1において、1はソースプログラム6を実行するソースプログラム実行部15によりプロセス間で実行されるメッセージ通信を取込んでプログラム実行順序を記録するプログラム実行順序記録部、2はこのプログラム実行順序が通信履歴として書込まれる履歴記憶部、3はこのソースプログラム実行により得られたデバッグ対象プロセスの出力値が正しいか誤っているかの情報を入力する正誤入力部、4はこの正誤入力部3より出力される正誤情報を記憶する正誤記憶部である。

【0012】また、5はデバッグ対象プロセス実行部16により取込まれたソースプログラム6からの情報と履歴記憶部2に記憶されたプロセス間の実行順序の履歴に基づいてデータの因果関係を解析し、デバッグ対象プロセスのデータ間の参照と定義の関係を含むデータ因果関係情報を作成するデータ因果関係作成部、7はこのデータ因果関係作成部5で作成されたデータ因果関係情報を記憶するデータ因果関係記憶部、8はこのデータ因果関係記憶部7に記憶されたデバッグ対象プロセスのデータ因果関係情報と正誤記憶部4に記憶された正誤情報を用いてプログラムエラーの可能性を計算し、プログラムエラーが含まれる可能性の高さを順位付けするエラー可能性予測部で、このエラー可能性予測部8で予測されたプログラムエラー予測性の順位はエラー可能性順位記憶部9に記憶される。

【0013】一方、10は履歴記憶部2に記憶されているプロセス間のプログラム実行順序の情報を取込んで同期事象であるか非同期事象であるかを判断して非同期事象部分を抽出する非同期事象抽出部で、この非同期事象抽出部10により抽出された非同期事象部分は非同期事象部分記憶部11に記憶される。また、12はエラー可能性順位記憶部9に記憶されているエラー可能性の順位と非同期事象部分記憶部11に記憶されている非同期事象部分を表示する表示部である。さらに、13は実行順序の入替え通信指定部14より表示部12に表示された表示内容に従ってプロセス間の実行順序を入替えるメッセージが入力されると共に、履歴記憶部2より実行順序を取込んでその実行順序の入替えの再実行指令を出力する実行順序入替え再実行部である。

【0014】次に上記のような構成された並行プログラムのデバッグ支援装置の作用について述べる。いま、例えば図2に示すような3つのプロセスX、Y、Zからなる並行プログラムがあり、デバッグ対象プロセスXが求める結果がK1、K2、K3の値であるとする。また、今回のプロセス間のプログラム実行順序が、図3のように行われたとする。このときのデバッグ装置の処理の流れを図4および図5に示し、図4におけるステップ104の詳細を図6に、図4におけるステップ105の詳細を図7乃至図9に示す。

【0015】まず、図4において、ステップ101により実行が開始されると、ソースプログラム実行部15によりソースプログラム6の情報に基づいて各プロセスを実行し、そのときのプロセス間のプログラム実行順序の監視により、そのプロセス間のプログラム実行順序がプロセス実行順序記録部1に取込まれる。このプロセス実行順序記録部1に取込まれたプログラム実行順序は、履歴記憶部2に履歴データとして格納される。図10は履歴記憶部2に格納された履歴データを示すものである。次にステップ102によってデバッグ対象プロセスを決める。図2の並行プログラムでは、出力結果がK1、K2、K

3の値なので、ここではデバッグ対象プロセスをプロセスXとする。次にステップ103ではプロセス間のプログラム実行により得られるプログラム実行順序の出力結果から判断されるプロセスXのK1、K2、K3の値の正誤情報を正誤入力手段3より正誤記憶部4に入力する。ここでは、説明の理解を容易にするためKの中身は図11に示すようにK1、K2の値が誤りで、K3の値が正しいものとして扱う。

【0016】次にステップ104では、デバッグ対象プロセス実行部16の実行によりデータ因果関係作成部5にソースプログラム6からの情報および履歴記憶部2に記憶されているプログラム実行順序の履歴が取込まれると、このデータ因果関係作成部5ではこれらの情報に基づいてデータの因果関係を解析し、デバッグ対象プロセスのデータ間の参照と定義の関係を含むデータ因果関係の作成を行い、その情報をデータ因果関係記憶部7に格納する。ここでは、図2に示すような並行プログラムの場合を例としているので、図12に示すようなデータ因果関係が作成される。

【0017】ここで、図6に示すフローチャートによりデータ因果関係の作成手順について詳細に述べる。図6において、ステップ401ではデバッグ対象プロセス実行部16によりプロセスXのみの再実行を行い、履歴記憶部2に格納されているプロセス間のプログラム実行順序の履歴とソースプログラム6からの情報をデータ因果関係作成部5に取込んで、パスを決める。この再実行により図2のX2行の分岐文では(2)が、X5行の分岐文では(4)がとられたとする。図2のプロセスXの中では、X1、X3、X4、X6、X8行でA、B、Cの値を受信し、それらをもとにV、W、K1、K2、K3を計算して出力しているので、図6のステップ402と403によりA、B、Cの値が入力変数ノードとして、V、Wの値が中間ノードとして、またK1、K2、K3の値が出力ノードとしてデータ因果関係記憶部7に記憶される。そして、ステップ404によりデータ因果関係記憶部7にあるノードをつないでいく。これを図2のプロセスXについてみると、X5行の分岐文では(2)を通り、AからK1の値に影響を与えているので、AからK1をつなぐ。X5行の分岐文では(4)を通るため、A、BがVに影響を与える。そこで、AからVとBからVをつなぐ。X7行ではWの決定にBが影響を与えているので、BからWをつなぐ。X9行ではVからK2、X10ではWとCからK3をつなぐ。このように各ノードどうしをつないでいくことにより、図12のデータ因果関係図が作成できる。

【0018】再び図4に戻り、ステップ105ではいままでの処理で得た正誤記憶部4にある正誤表(図11)とデータ因果関係記憶部7にあるデータ因果関係図(図12)を用いてエラー可能性予測部8によってどの受信動作によるメッセージの入力にエラーが含まれているかど

うかを可能性の高さで順位づけ、エラー可能性順位記憶部9に図13に示すようなエラー順位表を格納する。

【0019】ここで、エラー可能性予測部8でのエラー可能性予測と順位付けの処理の流れについて図7乃至図9により詳細に述べる。まず、図7のステップ501によりエラー順位表の変数名にデータ因果関係図のノード名を登録し、ステップ502で図11の正誤表から1つの変数の正誤を読み出し、ステップ503で正しいか否かを判定し、正しければステップ504にて点数を+1に、正しくなければステップ505にて点数を-1として出力変数K1、K2、K3に点数-1、-1、1が代入され、ステップ506にて遷移数が確定するとフラグが1にセットされる。そして、ステップ507にてすべての出力変数について終了したか否かが判定され、終了したことが判定されるとステップ508に進む。次にこのステップ508により中間変数Vが選ばれると、図8のステップ509により1個前のノードに確定フラグが立っているか否かを判定する。すなわち、図12のデータ因果関係図を参照すると、VからK2に線が引かれているので、これを辿り1つ前のノードであるK2に到達し、K2は確定フラグが立っているので、ステップ510によりこの点数と遷移数をVの点数と遷移数に加算する。これにより、Vは-1、1となる。その後辿ったことのない線は存在しないので、ステップ514によりVに確定フラグが立つ。

【0020】次に図7のステップ508によりBが選ばれると、BからVへの線を辿り、Vに確定フラグが立っていることから、Bの点数と遷移数に-1、1が加算される。次に再び図8のステップ509によりBからWへの線を辿ってWに到達するが、Wに確定フラグが立っていないので、ステップ511によりBは、点数、遷移数とも-1、1のままに次のWの値の計算にいく。同様にして、Wの点数、遷移数が1、1と確定した後、再びBのノードがステップ508で選ばれたとき、BからWを辿り、Bの-1、1の値にWの1、1が加算され、最終的にBの値は0、2になって確定する。

【0021】このようにして、A、Cの点数と遷移数も図7のステップ508から図8のステップ515によって確定する。すべての点数が確定した後は、ステップ516により入力変数A、B、Cに関しての順位づけが行われる。Aに関しては、点数-2、遷移数2なので絶対値が等しく点数が負なので、図9のステップ520によりエラーの可能性は最上位となる。Cはステップ521により絶対値が等しく、点数が正なので最下位となる。このようにして、エラーの可能性は高い順にA、B、Cとなり、最終的に図13に示すようなエラー順位表ができあがる。

【0022】次に再び図4に戻り、エラー可能性順位記憶部9にあるエラー順位表の表示とともに、その受信による入力が同期事象であるか、非同期事象であるかを合せて表示するために、ステップ106で非同期事象抽出部

10を用いて履歴記憶部2にある図10の履歴を解析する。これを見ると、図3のプロセス間のプログラム実行順序が分かり、図3の[1]のプロセスYからプロセスXのデータAの通信と[2]のプロセスZからプロセスXへのデータAの通信の間には同期がとられていないため、この2つのプログラム実行順序が確定できない非同期事象部分であることが分かる。同様にプロセスXあてのデータBの通信[3]と[4]の場合も非同期事象である。これにより、ステップ107では受信動作にエラーが含まれている可能性の高さを順位で示し、その受信動作が同期事象であるか、非同期事象であるかを示す。

【0023】ステップ107の表示に従ってユーザが非同期事象部分で、かつエラーの可能性の一番高いデータAを選ぶと、図5のステップ110によって実行順序入替え再実行部13によりメッセージの到着順を図3における[1][2]の順序ではなく、[2][1]の順序に換えて再実行する。これにより、結果が正しい値に変わったときには、この非同期事象部分がエラーを引き起こした原因と考えられ、この部分に同期事象をいれるべきであることがわかる。結果に改善が見られない場合は、またエラーの可能性の順位に従って再実行をしてデバッグ支援を進める。

【0024】このように本実施例では、3つのプロセスX、Y、Zからなる並行プログラムによりメッセージ通信をしながら処理を進める計算機システムにおいて、予めソースプログラムの実行によりプロセス間のプログラム実行順序をプログラム実行順序記録部1に取込んでこれを履歴記憶部2に格納し、次にデバッグ対象プロセスの実行により履歴記憶部2に格納された実行順序の履歴とソースプログラム6からの情報をデータ因果関係作成部5に取込み、これらの情報に基づいてデータの因果関係を解析し、デバッグ対象プロセスのデータ間の参照と定義の関係を含むデータ因果関係を作成するようにしたので、このデータの因果関係からデバッグ対象プロセスに存するプログラムエラーにつながる非同期事象部分の解析を支援することが可能となる。

【0025】また上記構成に加えて、上記履歴記憶部2に格納されたプロセス間のプログラム実行順序の履歴を非同期事象抽出部10に取込んでプログラム実行順序が同期事象であるか非同期事象であるかを判断して非同期事象部分を抽出すると共に、デバッグ対象プロセスのプログラム実行により得られるプロセスXの出力値K1、K2、K3の結果に対して正しいか誤っているかの情報を正誤入力部3により入力してこれを一旦正誤記憶部4に格納しておき、この出力結果の正誤情報をエラー可能性予測部8に上記データ因果関係作成部5で作成されたデータの因果関係と共に入力して各メッセージ通信によって引き渡される値にエラーが含まれている可能性の強さを予測し、このエラーが含まれている可能性の強さを非同期事象抽出部10により抽出された非同期事象部分

と共に表示部12に表示するようにしたので、必要最小限の非同期事象部分に対する実行順序の入替えで非同期事象部分のデバッグ支援を対話的に効率よく行うことができる。

【0026】さらに、実行順序入替え指定部14より実行順序入替え再実行部13に表示部12に表示された表示内容に従って入替えるメッセージを指定することにより履歴記憶部2よりプロセス間のプログラム実行順序の履歴を取込んで実行順序入替えの再実行指令が出力されるので、全ての非同期事象部分を組合わせてテストする必要がなく、デバッグに要する時間および手間を大幅に短縮することができる。

【0027】

【発明の効果】以上の述べたように本発明によれば、デバッグ対象プロセスの非同期事象部分の全てに対して実行順序を入替えてテストすることなく、必要最小限の非同期事象部分の組合せでデバッグを対話的に効率良く行うことができる並行プログラムのデバッグ支援装置を提供できる。

【図面の簡単な説明】

【図1】本発明による並行プログラムのデバッグ支援装置の一実施例を示すブロック構成図。

【図2】3つのプロセスからなる並行プログラムの一例を示す図。

【図3】同実施例におけるプロセス間の実行順序を示す図。

【図4】同実施例の作用を説明するためのフローチャートを示す図。

【図5】図4に続くフローチャートを示す図。

【図6】図4のデータ因果関係作成ステップの詳細を示すフローチャートを示す図。

【図7】図4の非同期事象抽出ステップの詳細を示すフローチャートを示す図。

【図8】図7に続くフローチャートを示す図。

【図9】図8に続くフローチャートを示す図。

【図10】図1の履歴記憶部に格納されたプロセス間の実行順序の履歴の一例を示す図。

【図11】図1の正誤記憶部に格納された正誤表の一例を示す。

【図12】図1のデータ因果関係記憶部に格納されたデータ因果関係図の一例を示す図。

【図13】図1のエラー順位記憶部に格納されたエラー順位表の一例を示す図。

【符号の説明】

1……プロセス実行順序記録部、2……履歴記憶部、3……正誤入力部、4……正誤記憶部、5……データ因果関係作成部、6……ソースプログラム、7……データ因果関係記憶部、8……エラー可能性予測部、9……エラー可能性順位記憶部、10……非同期事象抽出部、11……非同期事象部の記憶部、12……表示部、13……

グ対象プログラム実行部。

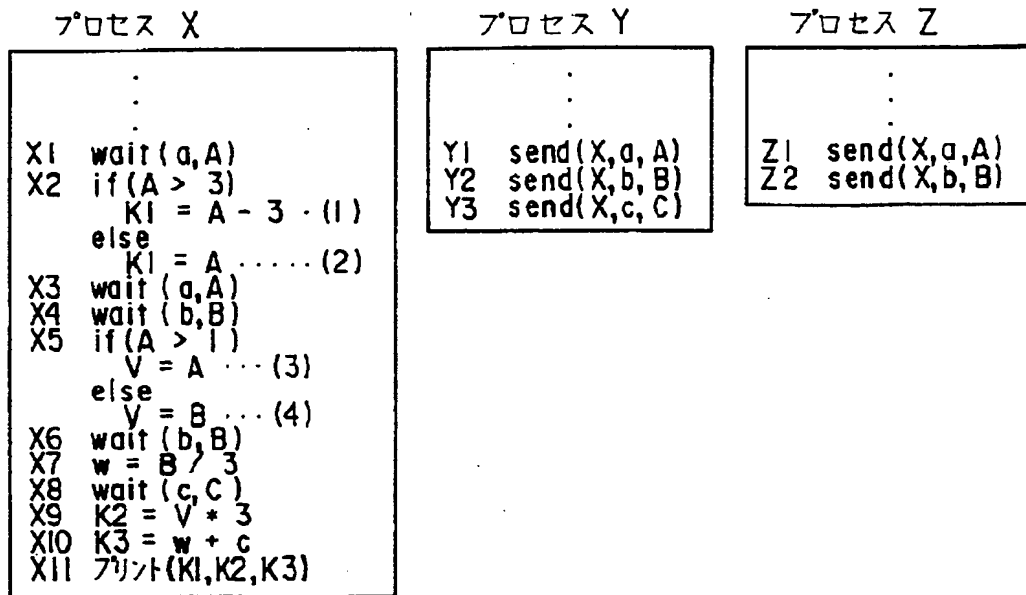
[illegible]

【圖 12】

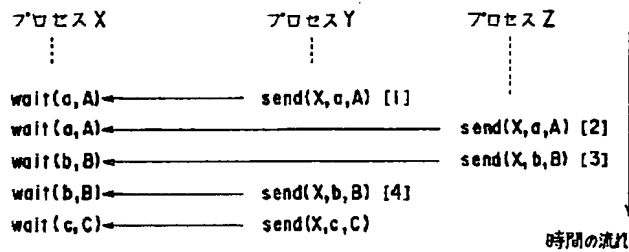
出力変数ノード 中間変数ノード 入力変数ノード

```
graph LR; A((A)) --> V((V)); B((B)) --> V; B --> W((W)); C((C)) --> W; V --> K1((K1)); V --> K2((K2)); W --> K2; W --> K3((K3));
```

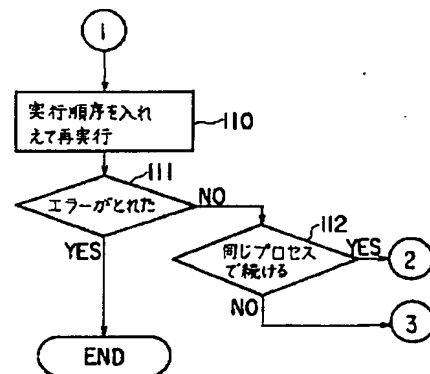

【図2】



【図3】



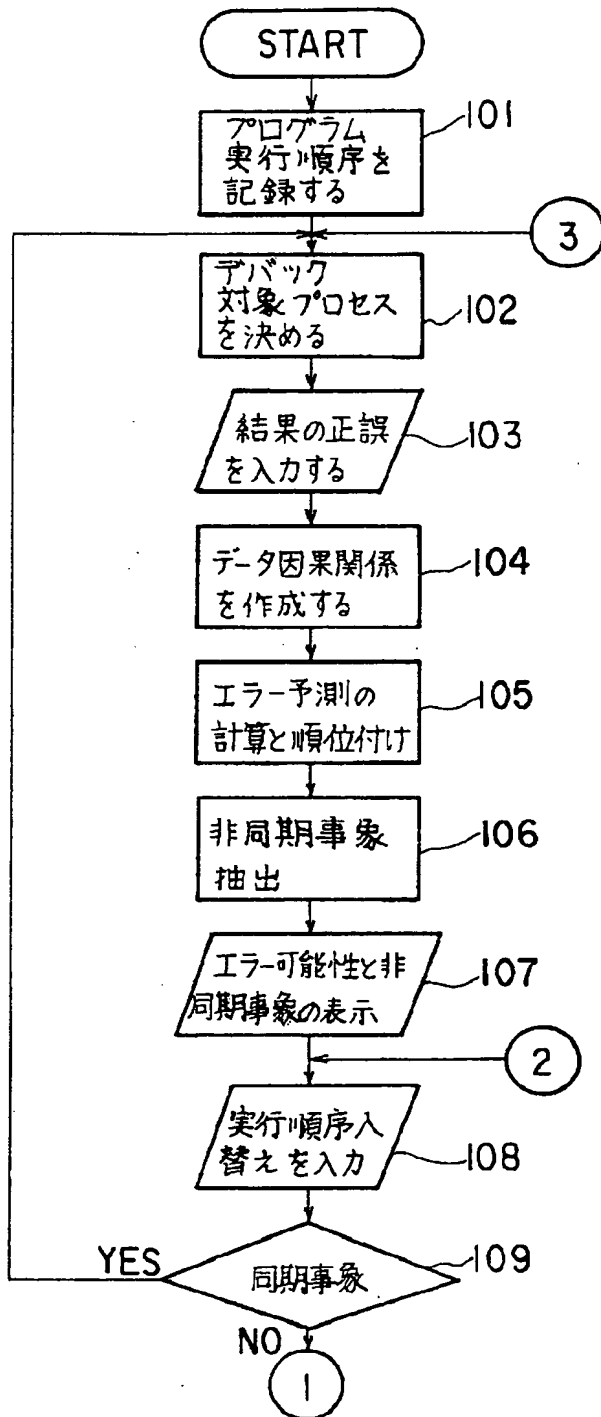
【図5】



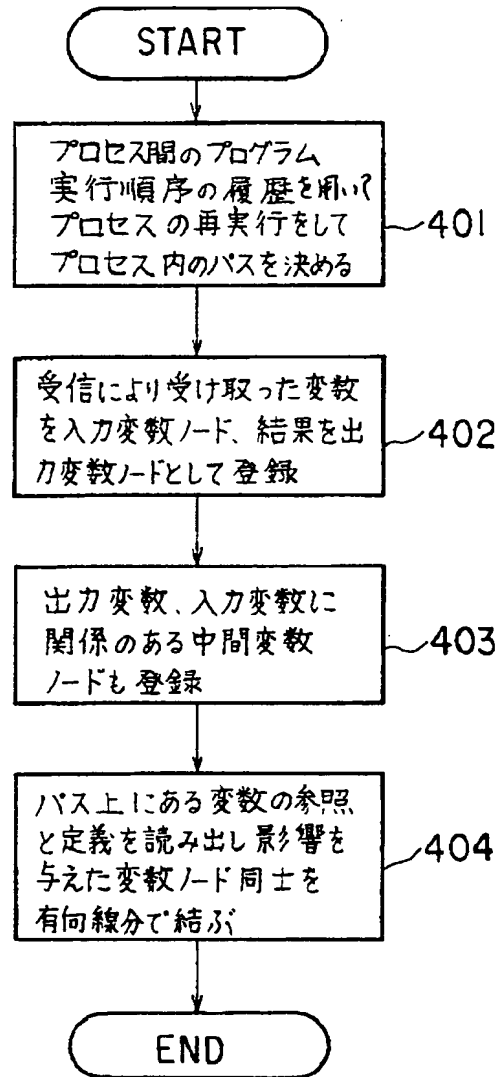
【図10】

通信の種類	受信プロセス名	送信プロセス名	データ
send	X	Y	A
wait	X	Y	A
send	X	Z	A
wait	X	Z	A
send	X	Z	B
send	X	Y	B
wait	X	Z	B
wait	X	Y	B
send	X	Y	C
wait	X	Y	C

【図4】



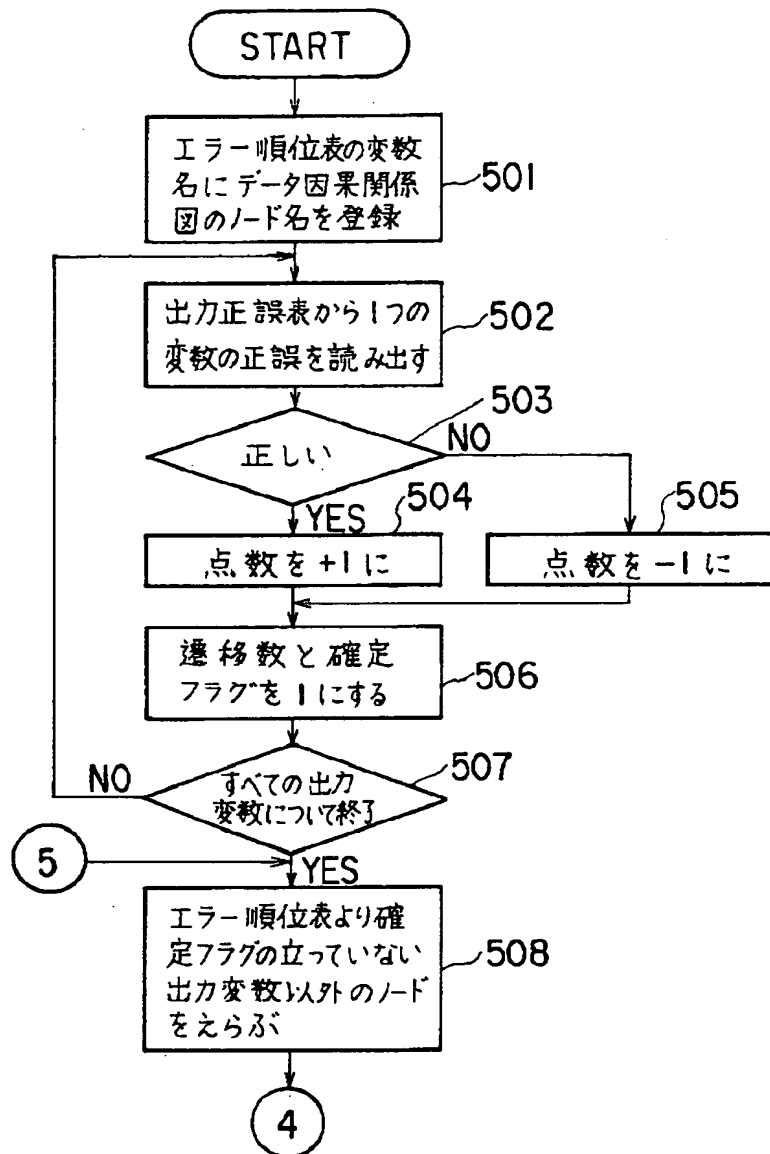
【図6】



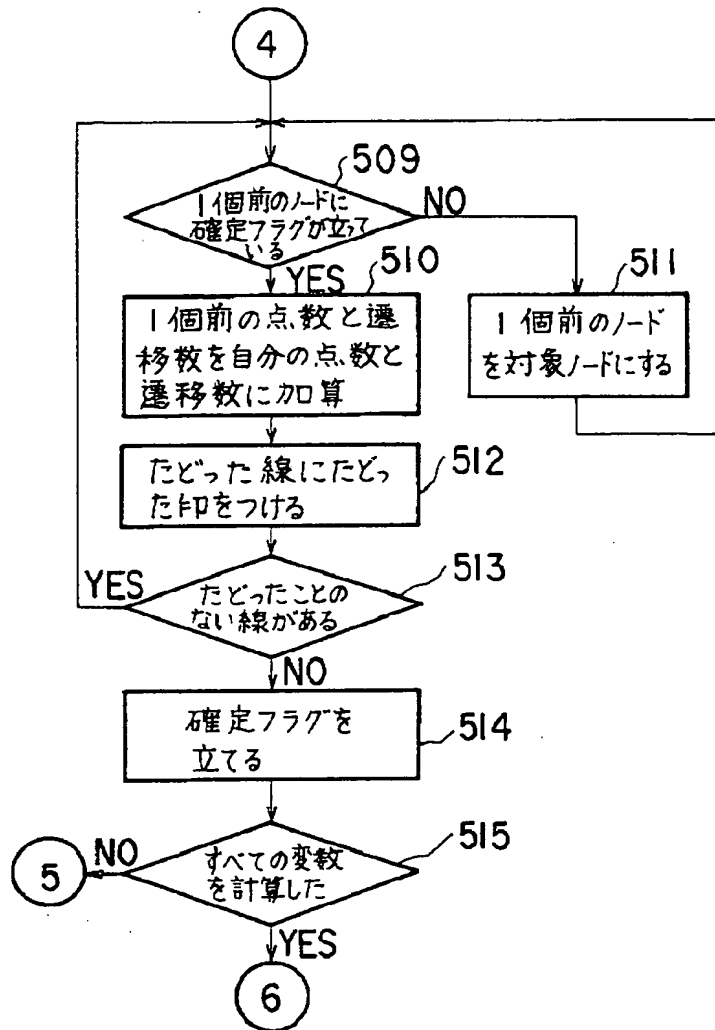
【図13】

変数名	K1	K2	K3	V	W	A	B	C
点数	-1	-1	1	-1	1	-2	0	1
遷移数	1	1	1	1	1	2	2	1
確定フラグ	1	1	1	1	1	1	1	1
順位						1	2	3

【図7】



【図8】



【図9】

